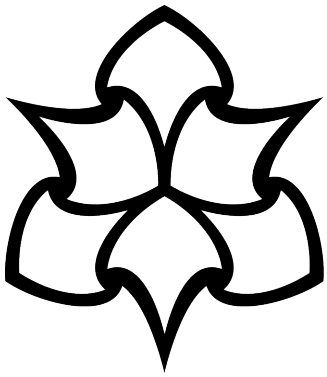

Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-019-01654-y

ORIGINAL RESEARCH

Strengthen user authentication on mobile devices by using user's touch dynamics pattern

Pin Shen Teh[1] · Ning Zhang[2] · Syh-Yuan Tan[3] · Qi Shi[4] · Wee How Khoh[5] · Raheel Nawaz[1]

Received: 11 June 2019 / Accepted: 12 December 2019

Abstract
Mobile devices, particularly the touch screen mobile devices, are increasingly used to store and access private and sensitive data or services, and this has led to an increased demand for more secure and usable security services, one of which is user authentication. Currently, mobile device authentication services mainly use a knowledge-based method, e.g. a PIN-based authentication method, and, in some cases, a fingerprint-based authentication method is also supported. The knowledge-based method is vulnerable to impersonation attacks, while the fingerprint-based method can be unreliable sometimes. To overcome these limitations and to make the authentication service more secure and reliable for touch screen mobile device users, we have investigated the use of touch dynamics biometrics as a mobile device authentication solution by designing, implementing and evaluating a touch dynamics authentication method. This paper describes the design, implementation, and evaluation of this method, the acquisition of raw touch dynamics data, the use of the raw data to obtain touch dynamics features, and the training of the features to build an authentication model for user identity verification. The evaluation results show that by integrating the touch dynamics authentication method into the PIN-based authentication method, the protection levels against impersonation attacks is greatly enhanced. For example, if a PIN is compromised, the success rate of an impersonation attempt is drastically reduced from 100% (if only a 4-digit PIN is used) to 9.9% (if both the PIN and the touch dynamics are used).

Keywords Mobile computing · User authentication · Behavioural biometrics · Touch dynamics

1 Introduction

Mobile devices have become a preferred gadget for users to access information and digital services, and stay connected. The increased usage and dependence on these devices also indicate that they increasingly process and store confidential and sensitive data. As more sensitive data are stored in, or accessible from, mobile devices, the risk and cost of losing these data are becoming higher. Therefore, more stringent security measures should be embedded into mobile devices. One of these measures is user authentication.

User authentication is the first line of defence in any computing system (platform or device). In a mobile device context, authentication is mostly achieved via a

✉ Pin Shen Teh
p.teh@mmu.ac.uk

Ning Zhang
ning.zhang-2@manchester.ac.uk

Syh-Yuan Tan
syh-yuan.tan@newcastle.ac.uk

Qi Shi
q.shi@ljmu.ac.uk

Wee How Khoh
r.nawaz@mmu.ac.uk

Raheel Nawaz
whkhoh@mmu.edu.my

[1] Department of Operations, Technology, Events and Hospitality Management, Manchester Metropolitan University, Manchester M15 6BH, UK

[2] School of Computer Science, University of Manchester, Manchester M13 9PL, UK

[3] School of Computing, Newcastle University, Newcastle upon Tyne NE4 5TG, UK

[4] Department of Computer Science, Liverpool John Moores University, Liverpool L3 3AF, UK

[5] Faculty of Information Science and Technology, Multimedia University, Malacca 75450, Malaysia

Published online: 24 December 2019

knowledge-based authentication method, and, in some cases, a fingerprint-based authentication method is also supported. With the knowledge-based authentication method, a user proves their identity by demonstrating the knowledge of a secret. The use of these secrets is vulnerable to certain security attacks such as smudge (Aviv et al. 2010), shoulder spoofing (Zakaria et al. 2011), and brute force (Owusu et al. 2012) attacks. The fingerprint-based authentication method is sometimes unreliable. For example, a sweaty, or dry, finger often leads to a false negative authentication result (Park et al. 2011). Therefore, how to make the authentication service secure and reliable in the presence of these threats and attacks for mobile users is a pressing task.

One of the possible measures to strengthen the security and reliability of the authentication service is to integrate a biometrics-based (e.g. touch dynamics) authentication method with a knowledge-based (e.g. PIN) authentication method to form a so-called two-factor authentication method. Touch dynamics refer to the digital signatures generated when a human interacts with a mobile device.

A touch dynamics authentication method can be implemented by employing sensors already available in most mobile phones, digital tablets, and other touch screen devices, making the implementation comparatively cheaper than other biometrics-based authentication methods, such as fingerprint and iris where specialised hardware is required. In addition, the acquisition of touch dynamics features is less sensitive to external factors such as lighting conditions and background noise levels, making it more usable and reliable in a mobile context. Also, touch dynamics features can be acquired whenever a user uses his/her devices, for example, during their normal (i.e. non-authentication related) input activities, requiring little extra interactions by the user. For these reasons, a touch dynamics authentication method is cheaper, more usable and reliable, and may be more acceptable to the general public than other biometrics-based authentication methods.

To investigate the feasibility and effectiveness of using touch dynamics biometrics as a mobile device authentication solution, we have designed and evaluated a touch dynamics authentication method. This paper reports our work in this regard. More specifically, it describes how a touch dynamics dataset is acquired, the way raw touch dynamics data are extracted from the dataset, what features and how they are extracted from the raw data. It also explains how we systematically analyse these features to select a subset of optimal features. It then describes the classification of the features to build authentication model, and the use of the model to authenticate a user. The paper also describes the experiments carried out to evaluate the performance of the touch dynamics authentication method and discusses the evaluation results obtained with different parameter value settings.

In detail, the structure of this paper is organised as follows. The next section discusses the related work. Section 3 describes a system architecture for our proposed touch dynamics authentication method. Sections 4, 5, 6 and 7, respectively, focuses on the designs and operations of core functional units of the architecture, i.e. the Raw Data Acquisition unit (Sect. 4), Feature Construction unit (Sect. 5), Model Training unit (Sect. 6), and Authentication Decision-Making unit (Sect. 7). Section 8 describes the performance evaluation methodology used. Section 9 analyses and discusses the evaluation results. Finally, Sect. 10 concludes the paper and outlines our future work.

2 Related work

This section critically analyses related work on touch dynamics. Depending on the input strings types used, the related work can broadly be classified into three groups: (i) work on numerical-based input strings, (ii) work on character-based input strings, and (iii) work on non-string inputs. As the scope of our work is in touch dynamics using numerical-based input strings, our literature critical analysis here will focus on (i). For details on related work in other groups, readers are referred to a recent literature survey of touch dynamics (Teh et al. 2016a). The work most relevant to ours has largely been focusing on studying the applicability of, or improving the performance in, using touch dynamics as a means of verifying subjects. We here discuss the most notable ones.

The work reported in (Vazquez et al. 2014) was carried out to test the applicability of verifying subjects based on touch dynamics using numerical-based input strings. In their experiments, some of the touch dynamics features were extracted by using the more sophisticated accelerometer and gyroscope sensors. By using an Euclidean Distance classifier, they obtained an accuracy performance of 20% equal error rate (EER) on a 4-digit PIN. Using more sophisticated sensors to extract some of the features means that this method is energy-consuming. For example, it consumes additional energy to obtain readings from an accelerometer sensor (Wang et al. 2017). In addition, relying on the use of these sensors also means that the method can only be deployed on mobile devices that are equipped with these sensors, limiting the scale of its deployment. This is also the case for the work reported in (Krombholz et al. 2016).

The authors in (Sen and Muralidharan 2014) also investigated the accuracy performance of touch dynamics using a 4-digit PIN. In their investigation, they recruited ten subjects. Each subject was asked to provide 100 input samples of a predefined PIN ("1593"). They used the Multi-Layer Perceptron classifier to classify the legitimate subjects, and these subjects can be correctly classified up to 86% of the

time. The result is encouraging, but to achieve the reported level of accuracy performance, they have made use of 100 input samples per subject to train the classifier. Acquiring such a large number of samples from the subjects is time-consuming and not always practical during an enrolment phase. It is not clear if the same level of accuracy performance could still be achieved when a smaller number of input samples are used. A similar case can be said for the work reported in papers (Coakley et al. 2016; Roh et al. 2016; Shen et al. 2016).

The paper (Tasia et al. 2014) reported an experiment carried out on a small number of input samples than those described above. Input PINs ranging from 4 to 8 digits in length were used. By using a simple statistical classifier, the authors were able to achieve an EER of 8.4%. The experiment was carried out on 100 subjects, more than many of the experiments reported in literature. For example, the experiments reported in papers (Amin et al. 2015; Buriro et al. 2015; Lee et al. 2016; Li et al. 2015; Praher and Sonntag 2016; Roh et al. 2016; Sen and Muralidharan 2014), were carried out on 15 or fewer subjects. Carrying out an experiment on a larger number of subjects allows us to draw more conclusive conclusions.

Zheng et al. (2014) conducted their experiments using 4-digit and 8-digit PINs. The authors employed a statistical classifier and obtained EER values of 3.65%, 6.96%, and 7.34% using three different 4-digit PIN numbers, "3244", "1111", and "5555", respectively. These results show that when a higher repetition of digits is used, the accuracy performance is reduced. They also compared the accuracy performances of two different 8-digit PINs (i.e. "12597384" and "12598416"). Surprisingly, for one of the 8-digit PIN ("12598416"), the EER value was 4.45%, marginally worse than the 4-digit PIN ("3244"). This finding is in contrary to previous studies (Chang et al. 2015; Praher and Sonntag 2016; Shen et al. 2016) which have suggested that longer input strings produce a better accuracy performance.

The work reported in (Chang et al. 2015) was somewhat unique. The authors proposed a method to allow subjects to change their PINs without rebuilding the authentication model. The subjects were asked to input ten different randomly selected 10-digit PINs. Based on the samples collected, they produced a table of all possible feature values for each digit. Using this method, they were able to achieve EER values of 23%, 21%, and 18% on three different PINs with the string lengths of 6, 8, and 10, respectively. However, the majority of the subjects taking part in this experiment were at the age of 17–20, it is not clear whether the experimental findings apply to other age groups.

The paper (Teh et al. 2016b) reported an experiment carried out to collect a touch dynamics dataset from a group of subjects with a wider age range than those described above. In their experiments, five timing feature and a spatial feature were extracted from the input samples of the dataset, and, for each extracted feature, a reference template is created. The authors applied three statistical methods, i.e. Gaussian Estimation, Z-Score and Standard Deviation Drift, to compare the likeliness of a test sample against a reference template. By using these three methods, they were able to achieve EER values of 8.55%, 9.30% and 8.92%, respectively, on a 4-digit PIN. These statistical methods were selected based on the notion that they are computationally less costly than the other machine learning methods, so the resulting authentication system could incurs less computational overhead, consumes less power and introduces less authentication delay. It would be interesting to investigate and compare the efficiency between different methods not only in terms of accuracy performance, but also the training and testing time they incur such as those investigated in our work.

Trojahn et al. (2013) reported an experiment carried out to investigate the accuracy performances of timing features extracted using different feature length values (i.e. 1-graph, 2-graph, and 3-graph) on a 17-digit PIN. The experimental results suggested that 1-graph achieves the best accuracy performance. A similar observation has also been reported in papers (Tasia et al. 2014; Zheng et al. 2014; Jain et al. 2014; Coakley et al. 2015). It would be interesting to investigate the accuracy performances of timing features extracted using larger feature lengths such as those investigated in our work.

More recently, Shen et al. (2016) investigated the accuracy performance of touch dynamics on 4-digit, 5-digit, and 6-digit PINs across different operational scenarios. Unlike other related work, they used only motion features extracted from accelerometer and gyroscope sensors. The authors pointed out that the raw data recorded by these sensors could not be used directly as features to build authentication models. To make the raw data useable as features, they computed a set of statistical metrics (min, max, mean, variance, etc.) from the raw data, and used the computed metrics as motion features. A similar method has also been used in other experiments such as (Buriro et al. 2015; Ho 2013; Zheng et al. 2014). By far, this method has only been used to extract motion features. It would be interesting to see how well this method works when used to extract other types of features, such as timing and spatial features investigated in our work. To investigate the effects of operational scenarios on accuracy performance, they designed three types of scenarios (i.e. hand-hold, table-hold, and walk-hold) for collecting subjects touch dynamics data. The results show that the hand-hold scenario achieves the best accuracy performance. A similar observation has also been reported in (Lee et al. 2016; Roh et al. 2016). Also, the results of these three pieces of work consistently show that the table-hold scenario achieves the lowest accuracy performance. This means that the features extracted from accelerometer and

gyroscope sensors may not be effective for user authentication purposes.

By far, the best accuracy performance was reported by (Wu and Chen 2015). The authors achieved an EER value of 0.56%. In this work, a two-class classification approach is used in building the authentication model, i.e. to build the model, samples from both legitimate and illegitimate subjects are used. This is also the case for the work reported in papers (Buriro et al. 2017; Ho 2013; Wu and Chen 2015). However, in real-life, as mobile devices are very much personal devices, illegitimate subject samples may not always be available. Therefore, this approach is less practical.

Table 1 summarises the related work discussed above and compares the related work against our work presented in this paper. This paper, in comparison with the related work discussed above, has presented a more systematic and comprehensive study of using touch dynamics biometric features for user authentication purposes. More specifically,

1. It proposes a touch dynamics authentication method, describing the design of a system architecture and its architectural units.
2. It gives a comprehensive description of the experiment carried out to acquire a touch dynamics dataset. Our experiment involves more subjects, acquires less number of samples per subject, and uses a device with a larger screen size than many other studies. The acquired dataset is made publically available, and unlike other public datasets (Antal and Nemes 2016; El-Abed et al. 2014), this dataset uses numerical-based input strings.
3. It discusses the extraction of not only a basic set of timing and spatial related features (FOF) from the dataset, but also an extended set of features (SOF) from the FOF features (related work has mainly focused on the extraction SOF from motion related features, rather than timing and spatial related features).
4. It investigates extensively how to make the most efficient use of touch dynamics biometrics in authenticating a user, in terms of optimising accuracy and efficiency performances. This includes the investigation and selection of a subset of optimal features, comparative studies of different timing feature lengths and different groups of classifier [one-class classifier (OCC) versus two-class classifier (TCC)], and the evaluation of the impacts of different parameter value settings on the performances.

3 Authentication system design

This section presents the threat model, deployment and working modes, and system architecture for the proposed touch dynamics authentication method.

Table 1 A summary of existing related works

Studies	Subject sizes	Input lengths	Sample sizes	Classifier groups	Device screen sizes	EERs (%)
(Sen and Muralidharan 2014)	10	4	100	TCC	3.7”	15.2
(Attaullah Buriro et al. 2015)	12	4	30	TCC	4.95”	97[a]
(Lee et al. 2016)	12	6	~117	TCC	5”	91.2[a]
(Wu and Chen 2015)	20	8	50	TCC	–	0.56
(Shen et al. 2016)	48	5	100	OCC	4.3”	9.74[b], 11.09[c]
					5.5”	5.01[b], 6.85[c]
					6”	4.53[b], 5.89[c]
(Zheng et al. 2014)	53	4	20	OCC	4.65”	3.65
	25	8				4.45
(Vazquez et al. 2014)	80	4,8	≥25	OCC	–	20
(A. Buriro et al. 2017)	95	8	30	TCC	–	0.01[b], 4[c]
(Tasia et al. 2014)	100	4–8	10	OCC	3.7”	8.4
(T.-Y. Chang et al. 2015)	100	6	5	OCC	3.7”	23
		8				21
		10				18
(Trojahn et al. 2013)	152	17	10	OCC	4.65”	4.19[b], 4.59[c]
This Study	150	4	10	OCC	10.1”	9.9
		16				7.1
		4		TCC		8.7
		16				5.3

[a]Accuracy; [b]FAR; [c]FRR

3.1 Threat model used

The threat model we use considers an attack scenario where an attacker or impersonator tries to gain access to a user's mobile device in an unauthorised manner. We make three assumptions in this scenario. First, the impersonator has physical access to the device. Second, the device is locked by a knowledge-based authentication method (e.g. a PIN). Third, the PIN is known to the impersonator. These assumptions are commonly made in biometrics related authentication evaluations (Stanciu et al. 2016). In this attack scenario, a knowledge-based authentication method will be compromised, and the impersonator will have unrestricted access to the mobile device.

Given this threat model, we demonstrate that, by integrating touch dynamics biometrics with a knowledge-based authentication method forming a so-called two-factor authentication solution, it is a lot harder to successfully bypass the authentication control, as, in this case, the impersonator would have to correctly input the pass code as well as to correctly reproduce the owner's touch dynamics biometrics. It should also be pointed out that we mainly consider authentication in the form of verification mode (i.e. to verify a claimed identity) instead of identification mode (i.e. to classify an unknown identity), as, in real life, mobile devices are rarely shared among multiple individuals, and most of the real-life application scenarios are in the verification mode.

3.2 Deployment and working modes

A touch dynamics authentication method can be deployed in one of the two modes, an identification mode and a verification mode. These modes function uniquely and serve different purposes and usage scenarios. The purpose served by the identification mode is to recognise or identify unknown identity. This mode is normally deployed for intrusion detections and forensic investigations. The purpose served by the verification mode, on the other hand, is to prove or verify a claimed identity. The authentication of a mobile user or a mobile device fits into this mode. Also, in real life, mobile devices are rarely shared among multiple individuals, and most of the real-life application scenarios are in the verification mode. For these reasons, we mainly consider authentication in the form of verification mode.

The verification mode can operate in two working modes, a static mode and a dynamic mode. In the static mode, a user is verified at the first instance of a user-to-system interaction. In the dynamic mode, a user may be verified at any instant of a user-to-system interaction or for every service access (i.e. continuously) throughout a service access session (in addition to the initial verification). The functions performed in both modes are complimentary. In other words, they can be deployed alongside each other to enhance the security of mobile devices or the security of service access using mobile devices. Our experiments are conducted under the assumption that the static working mode of the verification mode is used.

3.3 The architecture and its functional units

Figure 1 gives the system architecture for our touch dynamics authentication method. From the figure, it can be seen that the system architecture consists of six functional units, which run on a user's mobile device. The App Interface unit provides an input facility for users to input their touch dynamics data. The Data Storage unit (DSU) is a database used to store authentication model. The rest four units provide the core functions required to implement the touch dynamics authentication system.

The operation of a touch dynamics authentication system can broadly be captured in two phases, the enrolment phase and the verification phase. In the enrolment phase, the raw touch dynamics data of a subject (i.e. the owner of a mobile device) are acquired, processed, and transformed into an authentication model that is stored in the DSU. In the verification phase, the touch dynamics data of a test subject (i.e. a claimant) is compared against the authentication model retrieved from DSU to verify if the claimant is indeed whom he/she claims to be (i.e. the mobile device owner). Figure 1 indicates the two operational phases along with the units involved. In the next four sections, we describe the designs of the four core units and discuss the issues involved in more detail.

4 Raw data acquisition unit (RDAU)

RDAU is the first core functional unit of the proposed system architecture, responsible for extracting raw touch dynamics data from the subject's input samples. This section describes the design of this unit, giving detailed discussions with regard to how the raw data acquisition experiment is setup (Sect. 4.1), how the input samples are acquired (Sect. 4.2), how raw touch dynamics data are extracted from the input samples (Sect. 4.3), and how the raw data are processed into a proper format for further analysis (Sect. 4.4).

4.1 Experiment setup

The setup of an experiment carried out to acquire a touch dynamics dataset concerns a number of issues, namely, defining a data acquisition procedure, determining a physical environment, recruiting subjects, selecting a data acquisition device, and selecting input strings.

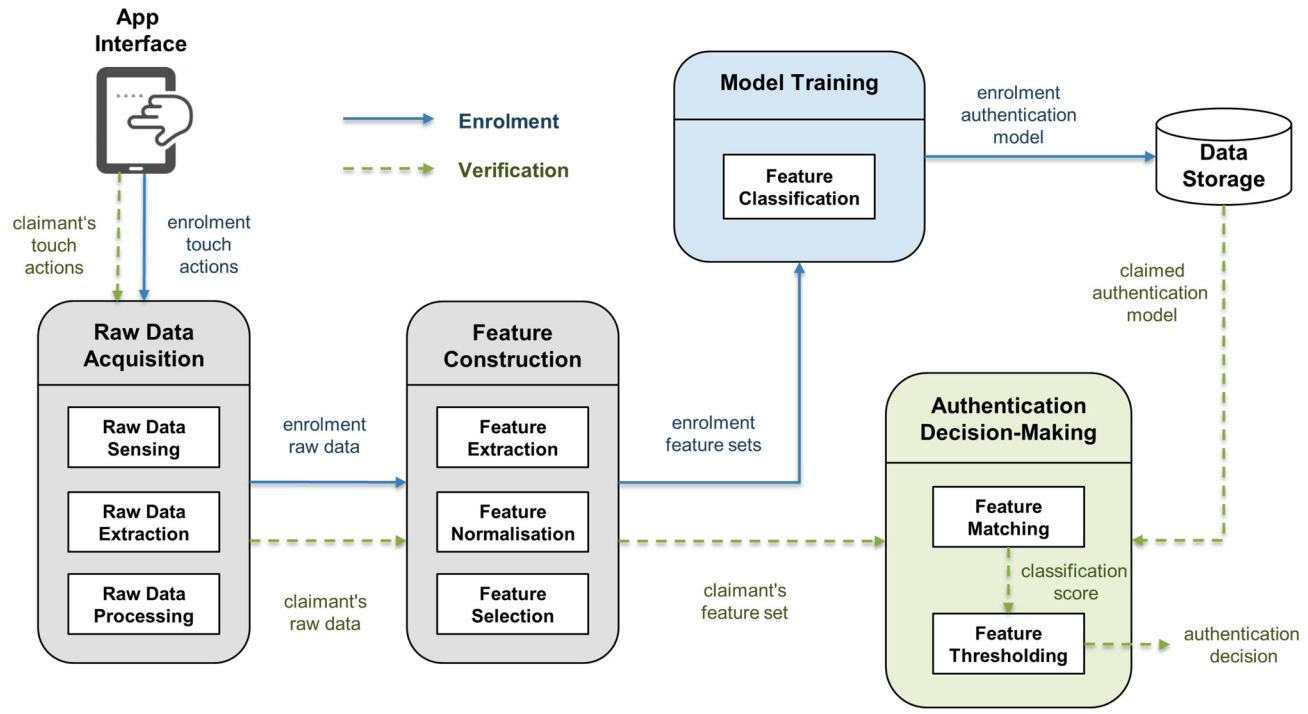

Fig. 1 The touch dynamics authentication system architecture

4.1.1 Defining a data acquisition procedure

To ensure that a subject's touch dynamics data are properly captured, a proper data acquisition procedure should be used. To design a proper data acquisition procedure, two issues should be considered.

The first issue is how to ensure data are acquired after a subject is in a stable state (i.e. after he/she is familiar, and comfortable, with the data acquisition procedure, device, and app). To address this issue, we have added a familiarisation time at the start of a data acquisition session. During this time, subjects were asked to familiarise themselves with the data acquisition procedure, device, and app. After the familiarisation time, the subjects were requested to complete two rounds of data acquisition, one using a 4-digit PIN and the other using a 16-digit PIN. In each round, the Data Acquisition App displays the required input PIN, and the subjects are asked to input the PIN. All subjects were required to repeat each PIN ten times.

The second issue is how to best capture intra-session variations for each subject. According to (Buschek et al. 2015), if all the samples from a subject are acquired in a single session, intra-session variations in the touch dynamics patterns of the subject may not be properly captured. To overcome this limitation, some authors (El-Abed et al. 2014; Tasia et al. 2014) suggest to break a single data acquisition session into multiple sub-sessions that are separated by some intervals, and then to combine the data acquired in the sub-sessions into a single set. However, this may means that the subject participation rate will be lower (De Luca et al. 2012). As a result, the sample size of the dataset may be reduced. To balance these considerations, i.e. to achieve a higher subject participation rate, while, at the same time, to better capture intra-session variations of the subjects, we have chosen to acquire data in a single session, but to add a familiarisation time to each subject's data acquisition session.

4.1.2 Determining a physical environment

To balance between preserving data quality and better capturing the subjects' touch dynamics patterns, we have established a semi-controlled data acquisition environment. We let subjects to choose their preferred locations where their touch dynamics data were acquired. The locations used included offices, homes, classrooms, inside vehicles, and public areas. To prevent inconsistent inputs or outliers mid-way into the data acquisition task, subjects were told to perform the required task continuously without breaks, avoid distractions, and stay focused while performing the task. To avoid distracting the subjects and to allow them to perform the data acquisition task naturally, we left the subjects alone when they are performing the task and monitor them from a safe distance. Also, to prevent subjects from deliberately altering the way they interacted with their devices and to capture their touch dynamics patterns in a

more natural manner, we explained the purpose of the study to the subjects only after they have completed the entire data acquisition session.

4.1.3 Recruiting subjects

In our experiments, 150 subjects were recruited. With regard to subject selections, three criteria are usually used, age, affiliation, and profession. Subjects from different age groups, affiliations, and professions may use their devices at different frequencies and/or have different levels of device familiarity. If subjects are not properly selected, the data acquired from the subjects may introduce unintended bias into experimental results, leading to inaccurate study results. To ensure study results are as unbiased as possible, and can be generalised to a wider population as much as possible, the subjects recruited in our experiments were from different age groups [< 20 (19%); 20–40 (44%); > 40 (37%)], had different device usage frequencies [rare (33%); average (21%); often (46%)], and were people from the general public working in different professions.

4.1.4 Selecting data acquisition devices

The device used in our experiments is a Samsung Galaxy Tab with a 10.1-inch screen, a 1 GHz dual-core processor, a 1-GB RAM, and it operates on Android 4.0.4. The device has a larger screen size than an average mobile phone and phablet (a hybrid of both a phone and tablet with a screen size between 5 and 7 inches (Pham et al. 2017). The decision for using a device with a large screen size is based on the following observations. Firstly, the average size of mobile devices in the market is getting bigger year after year (Ben Taylor 2014). Secondly, a majority of the devices used in literature have a screen size of approximately 5-inch or less (Aviv et al. 2012; Buschek et al. 2015; Coakley et al. 2016; Trojahn et al. 2013; Zheng et al. 2014). Thirdly, a subject's touch dynamics pattern may vary with different device sizes. This variation may be exploited to achieve a better accuracy performance.

4.1.5 Selecting input strings

What input strings should be used (and their lengths) is an important variable one should consider in the design of a data acquisition experiment. The PIN-based authentication method has so far been the most widely used authentication method for mobile devices (Aviv et al. 2017), so we have chosen this method as one of the two authentication methods to form our two-factor authentication method. We have selected two PIN inputs with different lengths. The first is a 4-digit PIN "5560" (hereafter referred to as 4D). This input string length is chosen to represent a PIN commonly used to unlock a mobile device or a debit/credit card. The second is a 16-digit PIN "1379666624680852" (hereafter referred to as 16D). This input string length is chosen to resemble a debit/credit card number commonly used when making a card-based online payment.

4.1.6 Our dataset

The entire dataset we have acquired consists of 3000 samples and 33,000 touch actions from 150 subjects. Each subject contributed a total of 20 samples (10 for the 4D string and 10 for the 16D string) from 220 touch actions (50 for the 4D string and 170 for the 16D string). The dataset is available to download at https://goo.gl/sNACU8.

4.2 Raw data sensing

Raw data sensing is the first process in RDAU. This is a process that obtains a subject's raw touch dynamics data during subject-to-device interactions. Raw touch dynamics data are normally acquired using specially developed App. The App was developed using Java and Android Application Programming Interface (API) Level 15. During a data acquisition session, the App prompts the subject to input a PIN by displaying a numeric keypad interface in a full-screen mode. For each key input of the PIN, four data values: (1) the key value; (2) the touch action type (i.e. finger pressing down on or releasing up from the key); (3) the touch action timestamp; and (4) the touch action pressure size.

Each touch action can be associated with a timestamp. The timestamp value represents the time when the action is taking place. A timestamp is recorded using the nanoTime() API function (Android Developers 2017a). It returns a time value with the highest timing precision (up to nanoseconds precision) that is available on the device. The pressure size can also be captured when a touch action is performed. The touch pressure size value represents the approximated size of the screen area being touched during a touch action. It is recorded using the getSize() API function (Android Developers 2017b), which returns a decimal value between 0 and 1.

4.3 Raw data extraction

Once input samples are acquired from each subject, raw touch dynamics data should be extracted from the samples. Each sample consists of m number of keys, and each key, $k_i, i \in \{1, 2, \ldots, m\}$, contains two touch actions: (i) the finger pressing down on the key, referred to as touch action press (TAP); and (ii) the finger releasing from the key, referred to as touch action release (TAR). The raw data are associated to these two actions, i.e. a timestamp, p, and a pressure size, ps, are recorded on each TAP action, and a timestamp, r, is

recorded on each TAR action. Figure 2 illustrates the raw data along with their respective touch actions.

4.4 Raw data processing

Once the raw touch dynamics data are obtained, the data should be processed into a format that is suitable for analysis and for subject verification. As discussed in Sect. 4.2, one of the raw data, the touch action timestamp, has a time value up to nanoseconds precision. This precision might be too high to properly capture a subject's touch action speed, as a human's touch action speed is usually at a slower pace than this order. If the timestamps are not set to an appropriate precision, the accuracy performance of the timing features, which are extracted from the timestamps, may be affected. So, it is important to choose an appropriate precision for the timestamps before extracting the timing features from them. Choosing an appropriate precision is done by using a scaling factor, α, i.e. a default timestamp, t, can be scaled by α to produce a scaled timestamp, $\hat{t}$, with the chosen precision, and this can be done by using the equation, $\hat{t} = t \times e^{-\alpha}$.

5 Feature Construction Unit (FCU)

FCU is responsible for extracting a subject's touch dynamics features from the subject's raw touch dynamics data. This section describes the design of this unit, giving detailed discussions with regard to the types of features that are extracted from the raw data and how the features are extracted (Sect. 5.1), the normalisation of the extracted features to the same value range (Sect. 5.2), and the selection of a subset of optimal features from the extracted features (Sect. 5.2).

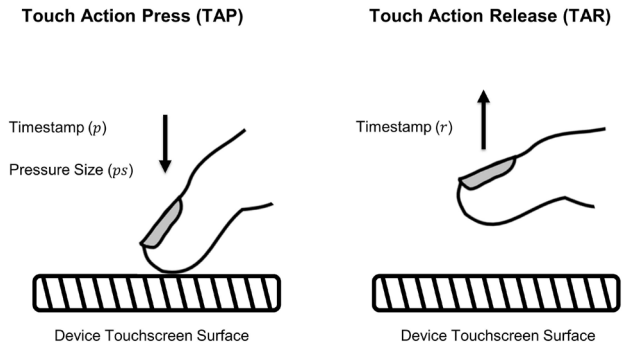

Fig. 2 Touch actions and their associated raw touch dynamics data

5.1 Feature extraction

In our design, two categories of features are extracted, first-order features (FOF) and second-order features (SOF). FOF features are a basic set of features extracted directly from the raw touch dynamics data, and SOF features are an extended set of features extracted from FOF features.

5.1.1 First-order features (FOF)

This section describes the process of extracting FOF features from a subject's raw touch dynamics data and constructing a cumulative FOF feature vector for the subject. For each subject, a number of FOF features are captured, one spatial feature and multiple timing features.

The pressure size (PS) is a spatial feature capturing the approximated size of the screen area being touched during a TAP. Each TAP is associated with a PS. A timing feature is an attribute capturing a time interval between two touch actions of one or more keys. Depending on how the intervals are measured, there are three types of timing features, i.e. dwell time (DT), flight time (FT), and input time (IT), and for FT, there are further four variants, i.e. FT1, FT2, FT3, and FT4. Timing features are extracted from timestamps. The descriptions of these timing features and the mathematical methods used to extract them from the respective timestamps are given in Table 2. The variable n in the equations refers to the timing feature length (to be discussed below) used to extract timing features and m refers to the number of keys in the input string.

A timing feature can be extracted at different feature lengths. A feature length is measured in terms of the number of graphs, i.e. the number of keys involved in each measurement. It is represented in the form of n-graphs, where n denotes the number of graphs. Figure 3 shows the different feature lengths for a given input string. The shortest feature length is 1-graph (or uni-graph), and the subsequent feature

Table 2 The descriptions and definitions of timing features

Features	Descriptions	Equations
DT	The interval between the TAP and TAR of a key	$dt_i = r_i - p_i$
FT1	The interval between the TAR of a key and the TAP of the next key	$ft1_i = p_{i+(n-1)} - r_i$
FT2	The interval between the TAR of a key and the TAR of the next key	$ft2_i = r_{i+(n-1)} - r_i$
FT3	The interval between the TAP of a key and the TAP of the next key	$ft3_i = p_{i+(n-1)} - p_i$
FT4	The interval between the TAP of a key and the TAR of the next key	$ft4_i = r_{i+(n-1)} - p_i$
IT	The interval between the TAP of the first key and the TAR of the last key	$it = r_m - p_1$

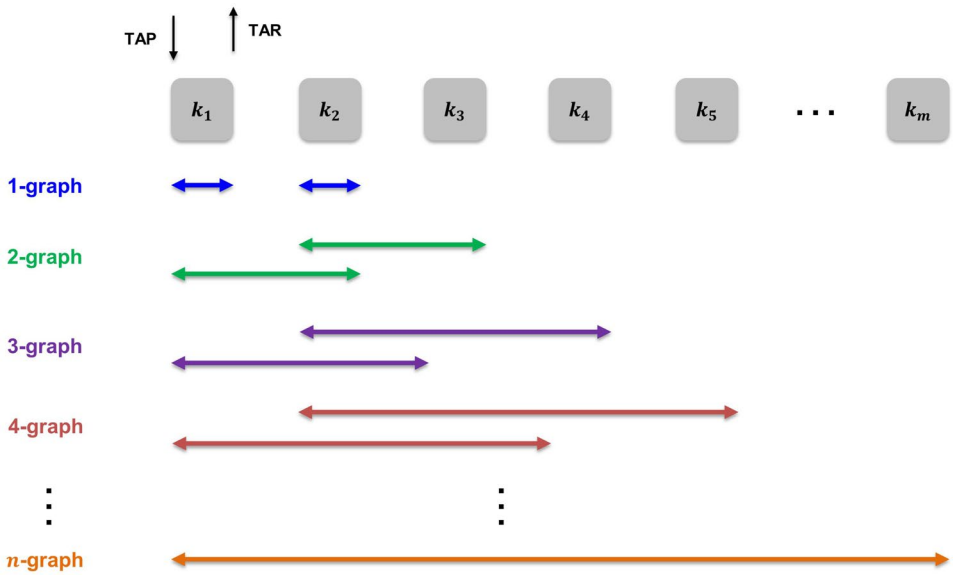

Fig. 3 The different feature length values

lengths are 2-graph (or di-graph), 3-graph (or tri-graph), and so on. With 1-graph, the only timing feature that can be extracted is DT. With a larger feature length, for example, when $n \geq 2$, the timing feature from two or more keys can be extracted, i.e. FT and IT. With regard to the feature length selection, in most of the experiments reported in literature, a feature length value of 2 is usually used (Trojahn et al. 2013). In our experiment, we have carried out a number of studies. In our study carried out to examine the effects of feature lengths on the accuracy performance of FT features, we have used different feature lengths. In our other studies, we mostly use a feature length value of 2.

Once FOF features, $f_i, i \in \{1, 2, \ldots, d\}$, are extracted from the raw data, they should be organised into the form of an FOF feature vector, i.e. $v_{FOF}^T = [f_1, f_2, \ldots, f_d]$, where T indicates a particular type of FOF feature and d refers to the feature dimension of v_{FOF}^T. When all the feature vectors are formed for a subject, a cumulative FOF feature vector, V_{FOF}, can be generated. This is done by concatenating the FOF feature vectors, i.e. $V_{FOF} = [v_{FOF}^{DT}, v_{FOF}^{FT1}, \ldots, v_{FOF}^{PS}]$.

5.1.2 Second-order features (SOF)

Some classification algorithms (or classifiers) perform better with a larger number of features (Ho 1998), so increasing the number of features used in training the classifier to generate an authentication model can improve the accuracy performance of the model. For this reason, we extract a new category of features, known as the second-order features (SOF), from FOF features, and use both of them in the training of authentication model. As discussed above, FOF features extracted from the raw touch dynamics data are organised into FOF feature vectors. For each of these vectors, a set of SOF features is extracted. The set consists of 19 features, and each feature represents a descriptive statistics metric of the FOF feature vector concerned. Descriptive statistics metrics are used to quantitatively summarise or describe a collection of data in a meaningful way (Prem 2016). The list of descriptive statistics metrics used in our experiment (with their corresponding feature identifiers in brackets) are: Minimum (mn), Maximum (mx), Arithmetic Mean (am), Quadratic Mean (qm), Harmonic Mean (hm), Geometric Mean (gm), Median (md), Range (rg), Variance (vr), Standard Deviation (sd), Skewness (sk), Kurtosis (ku), First Quartile (fq), Third Quartile (tq), Interquartile Range (ir), Mean Absolute Deviation (ma), Median Absolute Deviation (mi), Coefficient of Variation (cv), and Standard Error of Mean (se).

Similar to the case for FOF features, SOF features, $f_i, i \in \{mn, mx, \ldots, se\}$, once extracted from FOF features, should be organised into the form of a SOF feature vector, i.e. $v_{SOF}^T = [f_{mn}, f_{mx}, \ldots, f_{se}]$, where T indicates a particular type of FOF feature. When all the SOF feature vectors are formed for a subject, a cumulative SOF feature vector, V_{SOF}, can be generated. This is done by concatenating the SOF feature vectors, i.e. $V_{SOF} = [v_{SOF}^{DT}, v_{SOF}^{FT1}, \ldots, v_{SOF}^{PS}]$. Once both FOF and SOF cumulative feature vectors are formed for a subject, they should be combined into a form of a feature set, i.e. $\tau = \{V_{FOF}, V_{SOF}\}$. A feature set consists of the FOF and SOF features extracted from the raw touch dynamics data of a single input string of a subject.

5.2 Feature normalisation

If different features have different value ranges, the values of the features should be normalised. When features have different value ranges, they may have unbalanced weights at representing the structure of the data. Also, some algorithms

perform better and faster if the features have the same value range (Juszczak et al. 2002). For these reasons, the values of different features should be normalised to the same value range, this is usually done by using a process called feature normalisation.

In our experiments, we have conducted a feature normalisation process, and in this process, we have used a method called the min–max normalisation (Jain et al. 2005). With this method, the values of the features are scaled so that their ranges are confined to a predefined lower and upper boundary. Let X denotes a dataset of feature sets, $\tau_{ij}, i \in \{1, 2, \ldots a\}, j \in \{1, 2, \ldots d\}$, represented in the form of an $a - by - d$ matrix, where a refers the number of feature sets in the dataset and d refers to the feature dimensions of the feature sets. The matrix can be represented as:

$$X = \begin{matrix} \tau_{11} & \cdots & \tau_{1d} \\ \vdots & \ddots & \vdots \\ \tau_{a1} & \cdots & \tau_{ad} \end{matrix}$$

Then, the normalised dataset $\widehat{X}$ is obtained by using the following equation:

$$\widehat{X} = \frac{X_{ij} - min(X_j)}{max(X_j) - min(X_j)} \times (u - l) + l$$

5.3 Feature selection

Feature selection ensures an optimal set of features is used in training an authentication model to improve the accuracy performance of the model, while, at the same time, the cost incurred in training the model can be kept minimal. Typically, there are two sets of features, a preliminary feature set (PFS) and an optimal feature subset (OFS). PFS is a set of features that are directly extracted from raw touch dynamics data. OFS is a subset of features that are selected from the features in PFS. Feature selection involves two tasks, the selection of a feature selection method and the implementation of a feature selection process using the selected method.

We need a feature selection method that analyses the features in a PFS and selects an OFS from the PFS that satisfies two criteria. First, the OFS should contain the most relevant features. Second, the OFS should contain the least number of redundant features. In other words, the features in the OFS should have maximum relevance to the target variable (in our work, this is the identity of the subject, b), and, at the same time, have minimum redundancy amongst the features in the subset. In this work, we have chosen to use the minimum-redundancy-maximum-relevance (mRMR) method (Peng et al. 2005) as our feature selection method, as this method quantifies how well a feature, f_i, satisfies the two criteria mentioned above, and this is done by using a scoring metric,G. This metric is defined as:

$$G(f_i) = MI(f_i, b) - \frac{\sum_{j \in F} MI(f_i, f_j)}{|F|}$$

where the first part of the equation measures the degree of relevance between f_i and b, and the second part measures the degree of redundancy between f_i and the other features in the PFS, F. The higher the value of G, the better f_i is at satisfying both criteria.

With regard to the second task, the feature selection process is based on the mRMR method described above. The mRMR method is implemented by using the FEAST toolbox (version 1.1.4) (Brown et al. 2012). The process consists of the following four steps:

Step 1: The feature values in the PFS are converted to discrete values using the histogram bin counts method (MathWorks 2016).
Step 2: Starting with the first feature, G is calculated and assigned to the feature. This step is repeated for each of the remaining features in the PFS.
Step 3: Based on G, the features are sorted into a list. The features with higher scores are ranked higher in the list.
Step 4: Finally, the features ranked at the top-z% in the list are selected as the OFS, where z refers to the feature selection size.

In our study carried out to examine the effects of feature selection sizes on the accuracy performance, we have used different feature selection sizes. However, in our other studies, we used a feature selection size of 20.

6 Model training unit (MTU)

MTU analyses the touch dynamics feature sets (also referred to as touch dynamics samples or samples) extracted by FCU and trains them to generate an authentication model. The generated model should uniquely represent the corresponding subject's touch dynamics pattern. Model training is carried out by using a process called feature classification. The input of this process is the extracted OFS features, the algorithm used is a classifier, and the output is an authentication model. Feature classification involves two tasks, the selection of classifier and the implementation of the selected classifier.

Depending on the data used, classifiers can be classified into two groups, one-class classifier (OCC) and two-class classifier (TCC). An OCC classifier only uses data from a single class (e.g. data from legitimate subject). Unlike an OCC classifier, a TCC classifier uses data from two classes (e.g. data from both legitimate and illegitimate subjects). In the mobile device context, obtaining two classes of data

with a similar size is not practical. This is due to the fact that a mobile device is rarely shared among multiple users. Also, sharing a passcode with others increases data privacy risks and is not a recommended practice. For these reasons, obtaining illegitimate subject data is not practical, and only the data from the legitimate subject are available for use to train the classifier. If this is the case, a TCC classifier may not perform well, as it requires data from two classes to train a model that separates the two classes apart (Bellinger et al. 2012). By contrast, an OCC classifier only needs data from one class to train a model, so in this case, the model training is not affected by any imbalanced data. Besides, the time taken by a TCC classifier to train a model is longer than that by an OCC classifier, as the former uses more data in training the model. Based on these considerations, we have chosen to use OCC classifier for feature classifications.

With regard to the second task, we have implemented both OCC and TCC classifiers. For feature classifications, we have implemented two OCC classifiers: (i) one-class k-nearest neighbour (OCKNN) (Tax 2001), and (ii) the support vector data description (SVDD) (Tax and Duin 2004). For our comparative study, i.e. the study we have carried out to examine the effectiveness of using OCC classifiers versus using TCC classifiers, we have implemented two TCC classifiers as well: (i) k-nearest neighbour (KNN), and (ii) support vector machine (SVM) (Chang and Lin 2011). The classifiers used in our experiments are implemented using the Matlab (version 8.5.0.197613) programming platform and two open source toolboxes. The OCC and TCC classifiers are implemented using the dd_tools toolbox (Tax 2015) and the PRTools toolbox (Duin and Pekalska 2015), respectively. The implementation of the classifiers involves two phases: (i) the training (or enrolment) phase, and (ii) the testing (or verification) phase. In the training phase, training samples are used to train a classifier to build an authentication model. The built model is stored in DSU. In the testing phase, a testing sample is compared against the stored model to generate a classification score. This score will then be used to make an authentication decision.

7 Authentication decision-making unit (ADMU)

ADMU makes an authentication decision, i.e. whether a testing sample matches with the authentication model of the owner of the device. The design of ADMU involves two processes, feature matching and feature thresholding. In the feature matching process, the testing sample acquired from an authentication attempt is matched against the stored model in DSU to obtain a classification score. In the thresholding process, the score is compared to a predefined threshold, and if the score is over the threshold, then the sample is classified as legitimate. Otherwise, the sample classified as illegitimate.

8 Performance evaluation methodology

This section describes how the accuracy performance of our touch dynamics authentication method is evaluated. It covers the evaluation method, procedure and metrics used.

8.1 Evaluation method

To perform the accuracy performance evaluation of our touch dynamics authentication methods, we should perform the following four tasks. Firstly, we classify subjects into two sets, one designated as legitimate subjects and the other as illegitimate subjects. Secondly, some of the touch dynamics samples acquired from these subjects are used as training samples, in which these samples are used by MTU to generate authentication models. Thirdly, some of the other samples are used as testing samples, in which these samples and the generated models are used by ADMU to make authentication decisions. Lastly, based on the decisions, the evaluation metrics values are calculated, which indicates the accuracy performance of the model.

We need to acquire four sets of samples: (i) legitimate training set, (ii) legitimate testing set, (iii) illegitimate training set, and (iv) illegitimate testing set. To acquire the first two sets, we simply split the samples of each subject into two subsets, one as legitimate training set, and the other as legitimate testing set. To acquire the third set, we assign the samples of each subject that are not used as testing samples as the illegitimate training samples for all other subjects. To acquire the fourth set, we assign the samples of each subject that are not used as training samples as the illegitimate testing samples for all other subjects.

Once all the four sets of samples are acquired, they are used to evaluate the accuracy performance of our touch dynamics authentication method. The samples in (i) and (iii) are used by MTU to train classifiers to generate authentication models. The samples in (ii) and (iv) and the generated models are used by ADMU to make authentication decisions. The decisions are then compared against the actual classes of the testing samples to formulate a false acceptance count and a false rejection count. If a legitimate testing sample is incorrectly classified as illegitimate, the false rejection count is incremented. If an illegitimate testing sample is incorrectly classified as legitimate, the false acceptance count is incremented. These counts are used to calculate the evaluation metrics values described in the next section. The performance evaluation method described above is implemented using the evaluation procedure summarised in Algorithm 1.

Algorithm 1. Evaluation procedure

Input: Dataset S with B number of subjects, $\{d_1, d_2, \ldots, d_B\}$, Classifier C, folds K
Output: Accuracy performance of the model P

for $b = 1$ to B **do**
  $S^+ \leftarrow$ initialise the legitimate subject samples $\{d_b\}$
  $S^- \leftarrow$ initialise the illegitimate subjects samples $S - \{d_b\}$
  Randomly split S^+ and S^- into K disjoint folds, $\{s_1^+, s_2^+, \ldots, s_K^+\}$ and $\{s_1^-, s_2^-, \ldots, s_K^-\}$
  for $k = 1$ to K **do**
    $S_{tr}^+ \leftarrow$ initialise the legitimate training samples, $S^+ - \{s_k^+\}$
    $S_{tr}^- \leftarrow$ initialise the illegitimate training samples, $S^- - \{s_k^-\}$
    $T_{tr} \leftarrow$ initialise the training set, $S_{tr}^+ + S_{tr}^-$
    $T_{ts} \leftarrow$ initialise the testing set, $\{s_k^+\} + \{s_k^-\}$
    if C is a OCC classifier **then**
      $T_{tr} \leftarrow T_{tr} - S_{tr}^-$
      $T_{ts} \leftarrow T_{ts} + S_{tr}^-$
    end if
    Train C on T_{tr} to build model M_b
    $P_{fd} \leftarrow P_{fd} +$ (test the accuracy performance of M_b on T_{ts})
  end for
$P_{sb} \leftarrow P_{sb} + P_{fd}/K$
end for
$P \leftarrow P_{sb}/B$

8.2 Evaluation metrics

To evaluate the accuracy performance of the authentication model, three evaluation metrics are used, the False Rejection Rate (FRR), the False Acceptance Rate (FAR) and the Equal Error Rate (EER). FRR and FAR are also used to plot the Detection Error Trade-off (DET) curve (Martin et al. 1997), which is used to evaluate and compare the accuracy performances of different models in a graphical representation form.

FRR is calculated as the ratio of the false rejection count and the total number of legitimate testing samples. FAR is calculated as the ratio of the false acceptance count and the total number of illegitimate testing samples. EER is a single-number accuracy performance metric, which is calculated by averaging the FRR and FAR values with the condition that the absolute value of the difference between FRR and FAR is minimal (Chen 2003). Typically, the lower the FRR and FAR values, the lower the EER value. A lower EER value indicates a better accuracy performance of the model.

To plot the DET curve of a model, a set of FRR and FAR values of the model is needed. The values are obtained by setting the threshold to different values. The curve is formed by plotting the FRR values on the y-axis and the FAR values on the x-axis. The closer the curve to the bottom left corner, the better the accuracy performance of the model.

9 Performance evaluation results and analysis

This section describes the experiments carried out to evaluate the performance of the touch dynamics authentication method and discusses the evaluation results obtained. The results are presented in the following order, the evaluation of, RDAU (Sect. 9.1), FCU (Sect. 9.2), MTU (Sect. 9.3), and the authentication architecture as a whole (Sect. 9.4). The experiments were conducted using the evaluation methodology described in Sect. 8. Unless otherwise stated, each experiment was repeated four times, each time using one of the four classifiers (discussed in Sect. 6) in turn, and the results reported were the average of the classifiers.

9.1 Evaluation of RDAU

The evaluation of RDAU is carried out with different value settings for two parameters, i.e. the scaling factor and input string lengths.

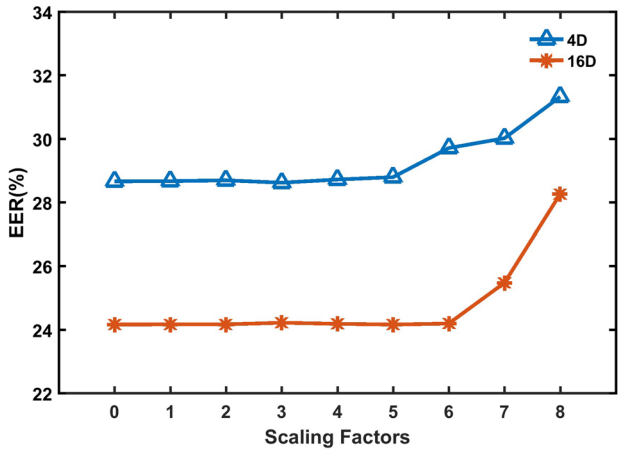

Fig. 4 EER values versus different scaling factor values

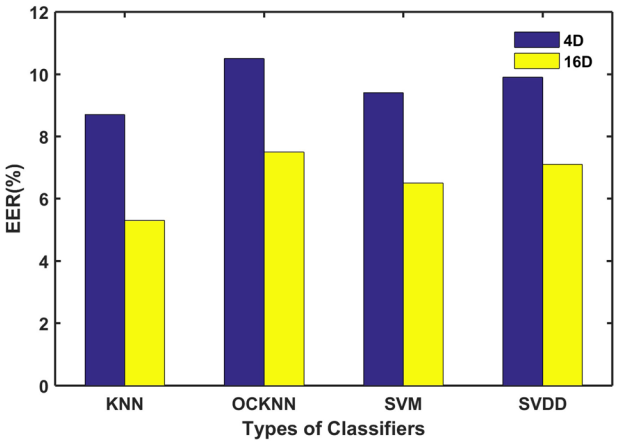

Fig. 5 EER values versus two different input string lengths and four classifiers

9.1.1 Scaling factor

The scaling factor is used to set the timestamps to different precision levels (as discussed in Sect. 4.4). To investigate the impact of using different scaling factor values, we have evaluated the accuracy performances, and the storage requirements of timing features by setting the scaling factor to be a range of values from 0 (no scaling) to 8 (maximum scaling) with an increment of 1. For each scaling factor value, we have extracted all the timing related FOF features from both the 4D string and the 16D string as the test cases.

Figure 4 shows the EER values versus different scaling factors, where two different input string lengths are considered. As can be seen from the figure, for both input string lengths, the EER values stay flat before a threshold value is reached. For the 4D string, this threshold value is 5, and for the 16D string, it is 6. Beyond the threshold values, the EER values increase steadily for the 4D string and sharply for the 16D string. These observations can be explained as follows. When timestamps are scaled using a smaller to a medium scaling factor, the timestamps have a proper precision to capture a human's touch action speed. As a result, the timing features, extracted from the timestamps, contain sufficient information to properly capture a human's touch dynamics pattern, resulting in stable EER values as observed. However, when using a very large scaling factor (7 or 8), the timestamps becomes smaller, and the timing features extracted from the timestamps contain less information, leading to an increase in the EER values. These observations indicate that, by scaling timestamps, the accuracy performance of the timing features extracted from the timestamps cannot be improved, but the use of an inadequate scaling factor value can worsen the accuracy performance.

It should also be emphasised that the use of different scaling factor values may affect the amount of storage space required to store the timing features. To investigate this effect further, we have recorded and compared the storage spaces used by the timing features extracted from timestamps with different scaling factor values. The results showed that the larger the scaling factor value, the less the amount of storage space that is used to store the timing features, though the level of reduction is small. For example, when the scaling factor value is set to a value between 0 and 5, the storage space used is reduced from 0.217 KB to 0.110 KB, and the reduction is only 0.107 KB. This level of reduction can be regarded as negligible, especially considering the fact that recently released mobile devices typically have a storage capacity of at least 8 GB to 32 GB (Bao et al. 2010), and the capacity is expected to increase in newer generation devices (Miluzzo et al. 2012).

Based on the above analysis, we can conclude that scaling timestamps does not bring much benefit. On the contrary, the scaling process introduces additional computational overhead. We, therefore, do not scale timestamps in our design. For some applications, for example, where the primary requirement is to minimise storage space, we recommend setting the scaling factor value to 5 (and not beyond).

9.1.2 Input string lengths

Using different input string lengths may also affect EER values. To examine the effect, we used two input strings with two different lengths, a 4D and a 16D string. For both input strings, FOF and SOF features were extracted. For each input string, we repeated the experiment four times, and for each time one of the four classifiers was used.

Figure 5 shows the EER values versus two input strings and four different classifiers. As shown in the figure, for all the classifiers, using the 16D string introduces a lower EER value, indicating that the longer the input string, the more accurate the authentication model. The reasons for

this are threefold. Firstly, the length of the 16D string is four times longer than that of the 4D string, and so is the number of features that are extracted from the 16D string. More features means more information about a subject's touch dynamics pattern can be captured, therefore a more accurate model can be built out of the features. Secondly, when the input string length increases, the number of possible chunk combinations also increases, and so is the ability to better capture a subject's touch dynamics pattern. Finally, when the input string length increases, the number of illegitimate features required to match that of a legitimate model will also increase, which means that the level of difficulty in impersonating a subject successfully also increases.

The above results have revealed a correlation between the input string length and security. The shorter the input string length, the lower the level of authentication accuracy, indicating a lower level of security. There is also a correlation between the input string length and usability. The longer the input string, the more the number of touch actions are required to complete the input of the string, thus the harder and slower it is for the users to memorise the string, indicating a lower level of usability. A similar correlation has also been reported in (Huh et al. 2015). In summary, the input string length influences the trade-off between security and usability. Therefore, in real-life applications, it should be chosen based on the security and usability requirements of the apps.

9.2 Evaluation of FCU

This section evaluates FCU with different parameter value settings, i.e. FOF features, FOF feature combinations, SOF versus FOF features, timing feature lengths, and feature selection sizes.

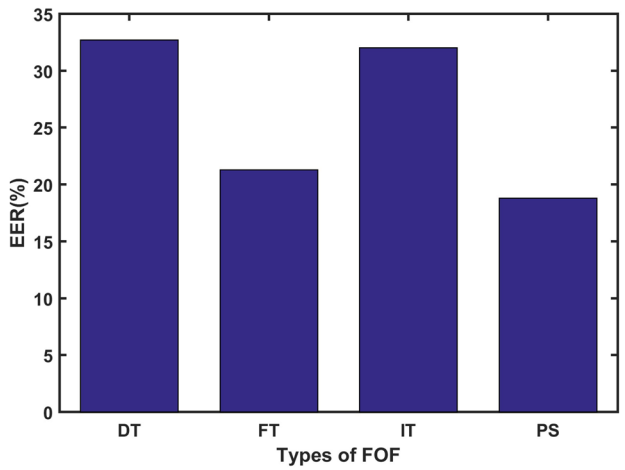

Fig. 6 EER values for different types of FOF

9.2.1 FOF features

There are four types of FOF (as discussed in Sect. 5.1.1). Each type of FOF captures a subject's touch dynamics pattern in a different way. To investigate the accuracy performance of the authentication method using different types of FOF, we have extracted all four types of FOF from the 4D string as the test case.

Figure 6 shows the EER values of the four types of FOF. The EER value of PS is the lowest amongst the four types, which means that the accuracy performance of PS is better than timing features. This result can be explained as follows. The PS values are determined by several factors such as: (i) the physical size of the fingertip used to perform a TAP; (ii) the amount of force exerted during a TAP; and (iii) the fingertip position or angle during a TAP. The combination of these factors creates a distinctive pattern, which allows PS to better capture each subject's touch dynamics pattern, and, as a result, achieves a higher level of accuracy.

With regard to the timing features, FT achieves the best accuracy performance in comparison with IT and DT. There are two reasons for this. Firstly, FT has a significantly larger feature dimensional space than IT and DT, and, as a result, more features are available for use in building the model that could better capture the subject's touch dynamics pattern. Secondly, FT has more control information and more discriminative properties than IT and DT, which may have originated from two sources. The first is the information embedded within the natural short pauses between different chunks. The second is the variability of chunk combinations. These can enable the classifiers to build models that better distinguish touch dynamics patterns from one subject to another. For these reasons, FT achieves a higher level of accuracy than DT.

A better way of understanding the accuracy performances achieved by different types of FOF is to visualise the feature values from different subjects graphically. Figure 7 shows the feature scatter plots of three types of FOF from three subjects. The subjects are randomly chosen. The x- and y-axis of each figure represents a type of FOF with the feature ID given in brackets. What is striking about the plots shown in the figure is that when PS is used (shown in Fig. 7c), the three subjects can be clearly distinguished or separated. However, this is not the case for FT (shown in Fig. 7b) and DT (shown in Fig. 7a). These observations are consistent with our discussions given above, i.e. PS achieves the best accuracy performance, which is followed by FT and, then, by DT.

9.2.2 FOF feature combinations

The results in Sect. 9.2.1 have shown that some types of FOF perform better than others. However, this does not mean

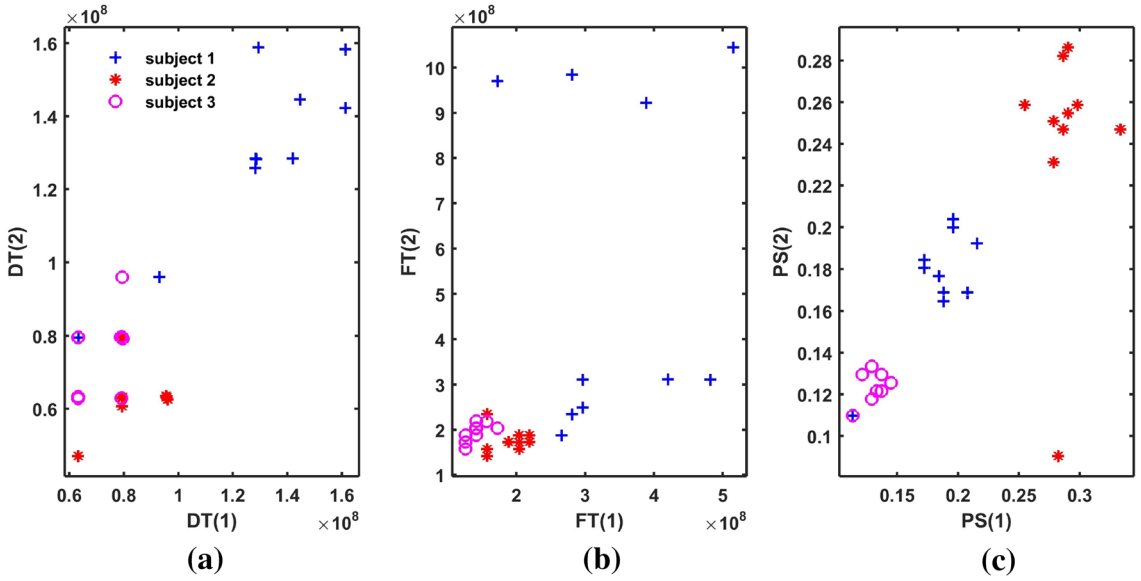

Fig. 7 Feature scatter plots of the three types of FOF: **a** DT, **b** FT, and **c** PS

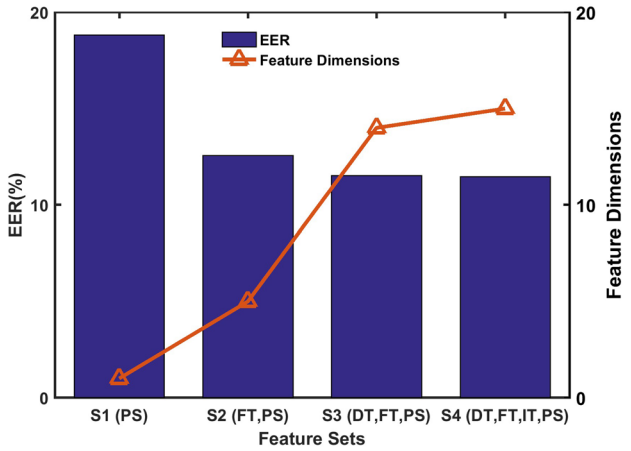

Fig. 8 EER values and feature dimensions for different feature sets

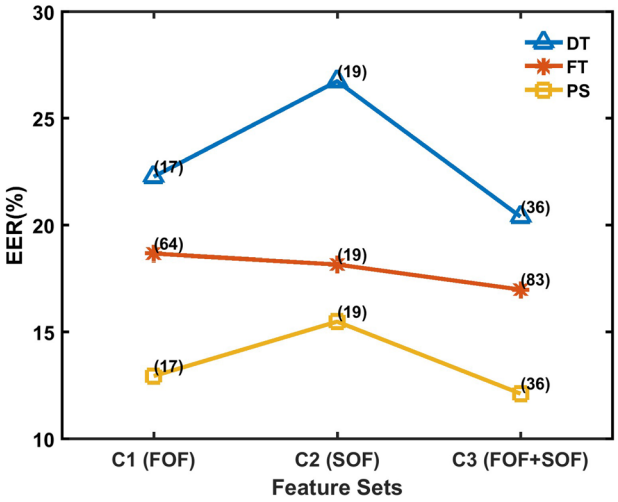

Fig. 9 EER values for different feature sets (with the feature dimensions in brackets)

that those under-performing types of FOF are not useful, as each type of FOF captures a different aspect of a subject's touch dynamics pattern. To investigate whether the accuracy performance of a model could be improved by combining multiple types of FOF, we formed 15 different feature set combinations by using the four types of FOF, and grouped them into four categories, i.e. S1(4), S2(6), S3(4), and S4(1). The numbers inside the brackets each indicate the number of feature set combinations in each category. The category number represents the number of types of FOF in each feature set of the corresponding category. Take the S3(4) category, for example, there are four feature set combinations, and each set is formed by using three types of FOF, i.e. {DT,FT,IT}, {DT,FT,PS}, {DT,IT,PS}, {FT,IT,PS}.

Figure 8 shows the EER values and the feature dimensions for different feature sets. For each category, only the best-performing feature set is shown. As can be seen from the figure, the larger the category number, the lower the EER value. In other words, the more types of FOF are combined, the better the accuracy performance we may achieve. For example, when S1 is used, the EER value is the highest (18.8%). However, when S2 is used, the EER value decreases to 12.55%, which is a marked drop, and when S3 and S4 are used, the EER value drops to 11.50 and 11.45%, respectively. These results can be explained as follows. When more types of FOF are combined, more features

are used in building the model, thus a more accurate model can be produced. However, the accuracy performance gain between the feature set S3 and feature set S4 is very small (only 0.5%). An explanation for this is that the features in S3 and S4 are very similar. The only difference is that S4 contains one additional feature, IT, which means that the features in S4 only capture a small amount of additional information in comparison with the features in S3, so there is only a minor improvement in the accuracy performance.

In summary, these experimental results suggest that the accuracy performance of the model can be improved by combining different types of FOF, and the combination of low-/medium-performing types of FOF can achieve a higher level of accuracy than the best-performing type of FOF when used individually.

9.2.3 SOF versus FOF features

SOF features are extracted from FOF features. To study the effectiveness of the SOF features, we have used three categories of features: (i) C1, containing the FOF features; (ii) C2, containing the SOF features that are extracted from the features in C1; and (iii) C3, containing the features in C1 and C2. Each category consists of three feature sets formed by using DT, FT, and PS, respectively. The features in the sets are extracted from the 16D string.

Figure 9 shows the EER values of the feature sets of three different categories. From the figure, we can see that, with the exception of C2 for FT, the EER values of C2 are higher than those of C1 in both cases of DT and PS. This may be due to that, as discussed in Sect. 5.1.2, the SOF features are descriptive statistics metrics, the fewer values available for use to generate the metrics, the less meaningful the metrics are at representing the trends of the values, and, as a result, achieves a lower level of accuracy. However, in the case of FT, the feature has a larger feature dimensional space than its counterparts, DT and PS, so more feature values are available when it is used to extract C2 (the SOF features) values. Therefore, we can get a better accuracy performance.

Based on the results, it seems that the accuracy performance of the authentication method with the use of C2 is not as good as that of C1, but this does not mean that they are not useful. C2 represents a subject's touch dynamics pattern in a different way from that of C1, and by combining the features in C1 and C2, the number of features available for use in training the model increases. As a result, the model has a better accuracy performance, meaning that the model could better capture the subject's touch dynamics pattern. For example, the EER value of C3 for DT (20.4%) is lower than the corresponding values of both C1 (22.28%) and C2 (26.75%). This is also true in the case of C3 for FT and for PS.

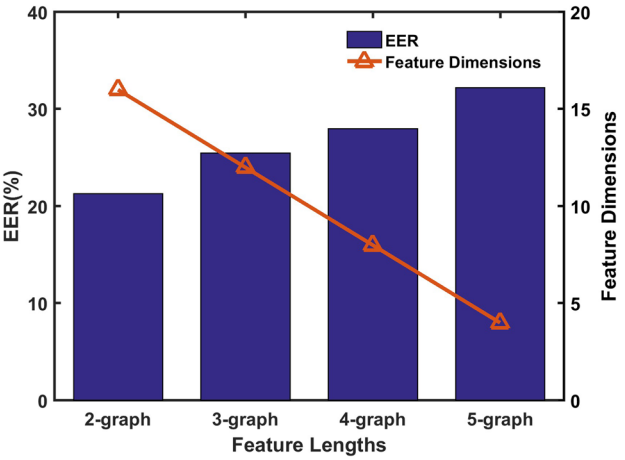

Fig. 10 EER values of FT versus different feature length values for the 4D string

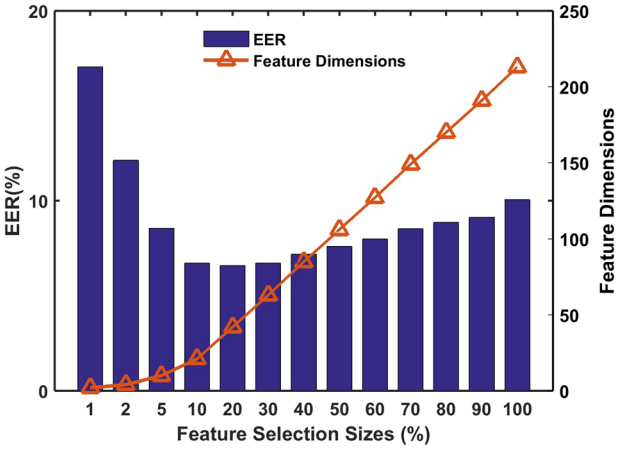

Fig. 11 EER values and feature dimensions versus the OFS sets at different feature selection sizes

9.2.4 Timing feature lengths

FT features can be extracted at different feature lengths. To study the correlation between the feature length and the accuracy performance, we set the values of the feature length to be a range of values from 2-graph to *n*-graph. The minimum and maximum value of the range respectively corresponds to the shortest and longest possible feature length used to extract FT features. In this test case, we have extracted the FT features from the 4D string.

Figure 10 shows the EER values and feature dimensions versus different values of the feature length. From the figure, it can be seen that there is a steady increase in the EER values as the value of the feature length increases. When the feature length is set to 2-graph, the EER value is the lowest at 21.28%. As the value of the feature length

increases, the EER value increases steadily, and when the feature length is set to 5-graph, the EER value reaches to the highest at 32.20%. These results indicate that the accuracy performance worsens when a longer feature length value is used. This could be due to the reason that timing features expressed using a longer feature length value contain a lower level of granularity, and thus capturing less information about a subject's touch dynamics pattern, leading to a lower level of accuracy. These results reflect those reported in (Giuffrida et al. 2014) where the authors also found that the accuracy performances of the features extracted with a shorter feature length are better than those with a longer feature length.

9.2.5 Feature selection sizes

In a feature selection process, the most important set of features are selected from a feature set, PFS, to form a subset of features, OFS, that could better represent the structure of the data. To investigate the benefit of using feature selection, we have used 13 different sets of OFS and compared their EER values. To form these sets of OFS, we first formed a PFS set consisted of the FOF and SOF features extracted from the 16D string. Then, we applied a feature selection process to the PFS set using 13 different feature selection sizes.

Figure 11 shows the EER values and feature dimensions versus the OFS sets with different feature selection sizes. As can be seen from the figure, the EER values are higher when the size is set to a very small or very large value. The EER values decrease sharply as the sizes increase from 1% to approximately 10%, remain mostly unchanged from 10 to 30%, before increasing slightly as the sizes increase toward 100%. These results can be explained as follows. When the smallest size (1%) is used, the OFS set has only 2 (out of 231) features. With such a small number of features available for training the model, the model does not have sufficient information to properly represent a subject's touch dynamics pattern, and, as a result, achieves the lowest level of accuracy (EER 17.05%). When larger sizes are used, the OFS sets have more features, and with more features available for training the model, the model could better represent a subject's touch dynamics pattern, and, as a result, the EER values decrease sharply. At a certain feature selection size, the EER value reverses the downward trend and starts gradually increase as the size increases further towards 100%. This change in trend usually happens when the feature dimension is relatively large, larger than the training sample size, such that the ability of the classifier to build an accurate model is reduced because of the large set of features (Gheyas and Smith 2010).

With the exception of the feature selection size of 1 and 2%, the accuracy performances for the other sizes are better than that of 100% (which is the case without feature selection or the use of the PFS set). This observation implies that, as long as the feature selection size is not set to an extremely low value, applying feature selection to the PFS set can improve the accuracy performance, and reduce the feature dimension of the PFS set, which leads to an increase in the efficiency and the robustness of the authentication method.

Table 3 EER, training, and testing time values of four classifiers

Classifiers	Classifier groups	EER (%)	Training times (unit)	Testing times (unit)
OCKNN	OCC	10.5	1	1
KNN	TCC	8.7	1	7
SVDD	OCC	9.9	3	1
SVM	TCC	9.4	22	2

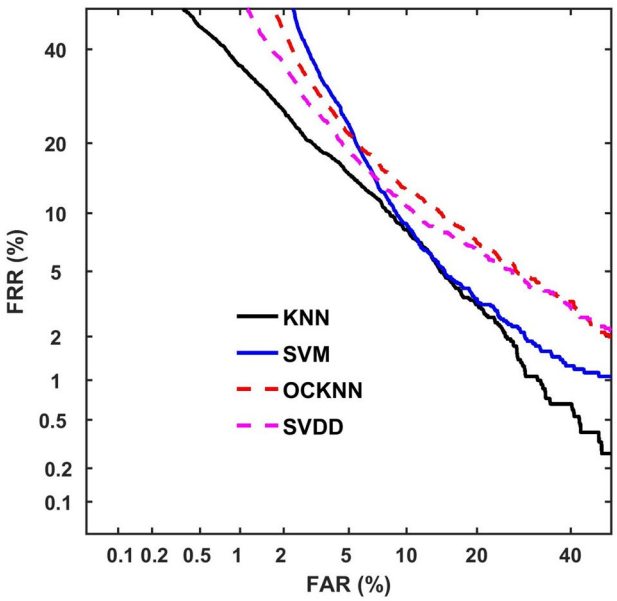

Fig. 12 DET curves of four different classifiers

9.3 Evaluation of MTU

The classifiers used in our experiments can be classified into two groups, OCC and TCC. The main difference between the two groups lies in the type of training samples they each use in building authentication models (as discussed in Sect. 6). Because of this, the models built by the two groups of classifiers differ in three attributes: (i) the accuracy performance, (ii) the training time, and (iii) the testing time. To evaluate and compare the two groups of classifiers in terms of these three attributes, we have chosen two classifiers for each group. For the OCC group, we have chosen OCKNN and

SVDD, and, for the TCC group, we have chosen KNN and SVM. The input to each of these classifiers is set to be the OFS set (with the feature selection size set to 20%) extracted from the 4D string.

Table 3 shows the EER values, training times, and testing times produced by using the classifiers, and Fig. 12 presents the DET curves of the classifiers. As shown in the table the EER values produced when using OCC are higher than those when using TCC. More specifically, the EER values when using OCKNN and SVDD are 10.5 and 9.9%, respectively, whereas the corresponding values when using KNN and SVM are, respectively, 8.7 and 9.4%. This indicates that the accuracy performances of the models built by OCC are lower than those by TCC. This may be due to the fact that, unlike OCC, the classifiers in TCC build the models with both legitimate and illegitimate samples, which means that the models can capture more information about the subjects' touch dynamics patterns, leading to more accurate models. However, it should be emphasised that the level of gain in the accuracy performance by using TCC is not significant. As shown in Fig. 12, the DET curves of OCC and TCC are somewhat close to each other.

Unlike the case for the accuracy performance, there is no clear correlation between a particular group of classifiers, OCC or TCC, and the model training time, rather the model training time appears to be classifier dependent. Among the four classifiers, SVM is significantly more expensive than the other three classifiers. The second most expensive classifier is SVDD, consuming 3 units of time against 1 by OCKNN and KNN. Two factors influence the model training time: (i) the nature (structure or approach) of a classifier, and (ii) the number of samples a classifier uses to train a model. It seems that the first factor plays a dominant role in model training time.

With regards to the model testing time, TCC classifiers are more expensive than OCC classifiers. KNN is the most expensive one among the four classifiers; seven times more expensive than OCKNN and SVDD. SVM is second most expensive, costing twice as much as OCKNN and SVDD. Similar to the case for the model training time, it seems that the nature of a classifier plays a dominant role in model testing time.

Based on the above results and discussions, particularly taking into consideration of the finding that, for a roughly similar level of accuracy performance, OCC classifiers are generally more efficient than TCC classifiers, both in terms of model training and testing times, and that, in a mobile device context, usually only the data from the owner of a device are available for use in training the classifier to build the authentication model, and performance and usability requirements are also important, we recommend the use of OCC classifiers in building an authentication model in this application context.

9.4 Evaluation of the authentication architecture

One potential application area of touch dynamics biometrics is user-to-mobile device authentication. For example, we could integrate a touch dynamics biometrics-based authentication method into an existing knowledge-based authentication method (e.g. a PIN) to produce a so-called two-factor authentication system. In this two-factor authentication system, the PIN serves one factor, and the touch dynamics serves the other factor. To evaluate the effectiveness and efficiency of this two-factor authentication method, we have compared two authentication systems: one using only a 4-digit PIN (denote as AS1), and the other using both a 4-digit PIN and the touch dynamics (denote as AS2).

In the evaluation, for both AS1 and AS2, we have used the assumption that the PIN has already been exposed to an impersonator. In this case, with AS1, the probability for the impersonator to successfully gain access to the user's device is 100%. On the contrary, with AS2, this probability is reduced to 9.9%, which is a significant reduction, indicating that the two-factor authentication method can achieve a significantly higher level of security in comparison with the single-factor method. Of course, there is a price to pay for using AS2; there is a non-zero FRR, which impedes usability. With this level of security enhancement offered by AS2, 1 out of 10 legitimate login attempts may be incorrectly rejected. With AS1, the FRR is zero, as none of the login attempts will be falsely rejected as long as the PIN is entered correctly.

The above evaluation results show that, with an additional authentication factor provided by using the touch dynamics biometrics, unauthorised accesses to mobile devices become harder, thus strengthening the security level of mobile devices. The results also show that, with the use of a touch dynamics based authentication method, there is a trade-off between security and usability. We leave the research question as for how to balance this trade-off to future investigation.

10 Conclusion and future work

This paper has investigated the feasibility and effectiveness of using touch dynamics biometrics for user authentication on mobile devices. To evaluate the effectiveness of this authentication method, we have acquired a comprehensive touch dynamics dataset. The method and process used to acquire this dataset have been clearly described and discussed. The paper has also extensively discussed how raw touch dynamics data may be extracted from the dataset, and how the raw data is processed into a proper format for feature extraction. In particular, it has explained that two types of features can be extracted, a basic set of features, FOF, that

can be extracted from the raw data, and an extended set of features, SOF, that can be extracted from FOF features. The paper then describes how the features may be analysed to select a subset of optimal features, and how the features may be classified using classifiers to build authentication models. Our experimental results show that the use of OCC classifiers is more efficient for roughly the same level of security than TCC classifiers, making the OCC-based classification method more practical in real-world applications.

Experiments were carried out to evaluate the performance of the touch dynamics authentication under various parameter settings. Experimental results showed that by integrating the touch dynamics authentication method into a 4-digit PIN-based authentication method, the success rate of an impersonation attempt is drastically reduced from 100% (if only the PIN is used) to 9.9% (if both the PIN and the touch dynamics are used). These results indicate that the idea of using touch dynamics biometrics to support user authentication in a mobile device or application context is feasible.

With regard to future work, we have identified the following issues that require further study. The first is to further investigate the scalability and viability of the touch dynamics authentication method by investigating the energy consumption and computational overhead introduced by the method on devices equipped with different battery capacity, CPU clock speed, and storage capacity, etc.

The second is to investigate whether there are additional features that could be captured to represent a subject's touch dynamics pattern. One way of doing this is by representing features in a different form, for example, by plotting the feature values in a graph, and then use the graph in the form of an image as the features to build authentication models. Alternatively, instead of searching for features manually, we could automate this process by using state-of-the-art deep learning techniques. These techniques have been widely used in the field of computer vision to automatically extract representative features from image related data (LeCun et al. 2010; Xu et al. 2014; Sun et al. 2014). Perhaps they can also be used to automatically find, from raw touch dynamics data, representative features to build accurate authentication models.

Thirdly, touch dynamics patterns can be affected by behavioural changes over time. These changes may cause trained model to deviate from the subject's most recent touch dynamics pattern. If this is the case, the accuracy performance of the model is reduced. In this regard, future research is necessary to explore how and to what extent these changes affect the accuracy performance, and what method may be used to accommodate these changes effectively.

Lastly, our dataset could be better with the use of data from more sensor modalities. We are working on using more recent device to capitalise on the wide range of available sensors such as orientation, inertial, accelerometer and gyroscope to collect a more comprehensive dataset. In addition, we aim to collect more samples from each subject so that there are sufficient samples for us to use more advanced algorithms, such as deep belief networks (Deng and Zhong 2013), to classify subject's touch dynamics pattern in an effective manner.

Data availability The dataset used to support the findings of this study is available to download at https://goo.gl/sNACU8.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this article.

References

Amin R, Gaber T, ElTaweel G (2015) Implicit authentication system for smartphones users based on touch data. In: Abraham A, Jiang XH, Snášel V, Pan J-S (eds) Intelligent data analysis and applications. Springer International Publishing, Berlin, pp 251–262. https://doi.org/10.1007/978-3-319-21206-7_22

Android developers (2017a) nanoTime—system. https://developer.android.com/reference/java/lang/System.html#nanoTime. Accessed 10 Dec 2017

Android developers (2017b) getSize—motionevent. https://developer.android.com/reference/android/view/MotionEvent.html#getSize(int). Accessed 10 Dec 2017

Antal M, Nemes L (2016) The MOBIKEY Keystroke Dynamics Password Database: Benchmark Results. In: Silhavy R, Senkerik R, Oplatkova ZK, Silhavy P, Prokopova Z (eds) software engineering perspectives and application in intelligent systems. Springer, Berlin, pp 35–46. https://doi.org/10.1007/978-3-319-33622-0_4

Aviv AJ, Gibson K, Mossop E, Blaze M, Smith JM. (2010). Smudge attacks on smartphone touch screens. In: Proceedings of the 4th USENIX conference on offensive technologies (pp 1–7). Berkeley, CA, USA: USENIX association. http://dl.acm.org/citation.cfm?id=1925004.1925009. Accessed 16 July 2015

Aviv AJ, Sapp B, Blaze M, Smith JM (2012) Practicality of accelerometer side channels on smartphones. In: Proceedings of the 28th annual computer security applications conference (pp 41–50). New York, NY, USA: ACM. https://doi.org/10.1145/2420950.2420957

Aviv AJ, Davin JT, Wolf F, Kuber R (2017) Towards baselines for shoulder surfing on mobile authentication. CoRR, abs/1709.04959. http://arxiv.org/abs/1709.04959

Bao X, Lee U, Rimac I, Choudhury RR (2010) DataSpotting: offloading cellular traffic via managed device-to-device data transfer at data spots. SIGMOBILE Mob. Comput Commun Rev. 14(3):37–39. https://doi.org/10.1145/1923641.1923655

Bellinger C, Sharma S, Japkowicz N (2012) One-class versus binary classification: which and when? In: 2012 11th international conference on machine learning and applications (ICMLA) (Vol 2, pp 102–106). Presented at the 2012 11th international conference on machine learning and applications (ICMLA). https://doi.org/10.1109/ICMLA.2012.212

Ben Taylor (2014, July 21) Why smartphone screens are getting bigger: Specs reveal a surprising story. http://www.pcworld.com/article/2455169/why-smartphone-screens-are-getting-bigger-specs-reveal-a-surprising-story.html. Accessed 16 Feb 2016

Brown G, Pocock A, Zhao M-J, Luján M (2012) Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. J Mach Learn Res. 13(1):27–66

Buriro Attaullah, Crispo B, Frari FD, Wrona K (2015) Touchstroke: smartphone user authentication based on touch-typing biometrics. In: Murino V, Puppo E, Sona D, Cristani M, Sansone C (eds), new trends in image analysis and processing—ICIAP 2015 workshops (pp 27–34). Springer International Publishing. http://link.springer.com/chapter/10.1007/978-3-319-23222-5_4. Accessed 25 August 2015

Buriro A, Gupta S, Crispo B (2017) Evaluation of motion-based touch-typing biometrics for online banking. In: 2017 international conference of the biometrics special interest group (BIOSIG) (pp 1–5). Presented at the 2017 international conference of the biometrics special interest group (BIOSIG). https://doi.org/10.23919/BIOSIG.2017.8053504

Buschek D, De Luca A, Alt F (2015) Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices. In proceedings of the 33rd annual ACM conference on human factors in computing systems

Chang C-C, Lin C-J (2011) LIBSVM a library for support vector machines. ACM Trans Intell Syst Technol 10(1145/1961189):1961199

Chang T-Y, Tsai C-J, Tsai W-J, Peng C-C, Wu H-S (2015) A changeable personal identification number-based keystroke dynamics authentication system on smart phones. Secur Commun Networ, n/a-n/a. https://doi.org/10.1002/sec.1265

Chen K (2003) Towards better making a decision in speaker verification. Pattern Recogn 36(2):329–346. https://doi.org/10.1016/S0031-3203(02)00034-1

Coakley Michael J, Monaco JV, Tappert CC (2015) Numeric-passcode keystroke biometric studies on smartphones. In: Proceedings of student-faculty research day (p B4.1-B4.6). Presented at the proceedings of student-faculty research day, Pace University. http://csis.pace.edu/~ctappert/srd2015/2015PDF/b4.pdf

Coakley MJ, Monaco JV, Tappert CC (2016) Keystroke biometric studies with short numeric input on smartphones. In: 2016 IEEE 8th international conference on biometrics theory, applications and systems (BTAS) (pp 1–6). presented at the 2016 IEEE 8th international conference on biometrics theory, applications and systems (BTAS). https://doi.org/10.1109/BTAS.2016.7791181

De Luca A, Hang A, Brudy F, Lindner C, Hussmann H (2012) Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In: proceedings of the SIGCHI conference on human factors in computing systems (pp 987–996). New York, NY, USA: ACM. https://doi.org/10.1145/2207676.2208544

Mendizabal-Vazquez I de, de Santos-Sierra D, Guerra-Casanova J, Sanchez-Avila C (2014) Supervised classification methods applied to keystroke dynamics through mobile devices. In: 2014 international carnahan conference on security technology (ICCST) (pp 1–6). Presented at the 2014 international carnahan conference on security technology (ICCST). https://doi.org/10.1109/CCST.2014.6987033

Deng Y, Zhong Y (2013a) Keystroke dynamics user authentication based on gaussian mixture model and deep belief nets. Research article, International scholarly research notices. https://doi.org/10.1155/2013/565183

Deng Y, Zhong Y (2013b) Keystroke dynamics user authentication based on gaussian mixture model and deep belief nets. Int Sch Res Notices. https://doi.org/10.1155/2013/565183

Duin RPW, Pekalska E (2015) PRTools 5.3.1, A Matlab Toolbox for Pattern Recognition

El-Abed M, Dafer M, El Khayat, R (2014) RHU Keystroke: A mobile-based benchmark for keystroke dynamics systems. In: 2014 international carnahan conference on security technology (ICCST) (pp 1–4). Presented at the 2014 international carnahan conference on security technology (ICCST). https://doi.org/10.1109/CCST.2014.6986984

Gheyas IA, Smith LS (2010) Feature subset selection in large dimensionality domains. Pattern Recogn 43(1):5–13. https://doi.org/10.1016/j.patcog.2009.06.009

Giuffrida C, Majdanik K, Conti M, Bos H (2014) I sensed it was you: authenticating mobile users with sensor-enhanced keystroke dynamics. In: S Dietrich (Ed.), detection of intrusions and malware, and vulnerability assessment (pp 92–111). springer international publishing. http://link.springer.com/chapter/10.1007/978-3-319-08509-8_6. Accessed 4 February 2015

Ho TK (1998) Nearest neighbors in random subspaces. In: Amin A, Dori D, Pudil P, Freeman H (eds) Advances in pattern recognition. Springer, Berlin, pp 640–648. https://doi.org/10.1007/BFb0033288

Ho G (2013) TapDynamics: strengthening user authentication on mobile phones with keystroke dynamics. Stanford University. http://cs229.stanford.edu/proj2013/Ho-TapDynamics.pdf. Accessed 18 June 2016

Huh JH, Kim H, Bobba RB, Bashir MN, Beznosov K (2015) On the memorability of system-generated PINs: can chunking help? In: eleventh symposium on usable privacy and security (SOUPS 2015) (pp 197–209). Ottawa: USENIX association. https://www.usenix.org/conference/soups2015/proceedings/presentation/huh. Accessed 12 Mar 2017

Jain A, Nandakumar K, Ross A (2005) Score normalization in multimodal biometric systems. Pattern Recogn 38(12):2270–2285. https://doi.org/10.1016/j.patcog.2005.01.012

Jain L, Monaco JV, Coakley MJ, Tappert CC (2014) Passcode keystroke biometric performance on smartphone touchscreens is superior to that on hardware keyboards. Inte J Res Comput Appl Inf Technol 2(4):29–33

Juszczak P, Tax D, Duin B (2002) Feature scaling in support vector data description. In: Proceedings ASCI (pp 95–102). Presented at the Conference. of the advanced school for computing and imaging

Krombholz K, Hupperich T, Holz T (2016) Use the force: evaluating force-sensitive authentication for mobile devices. In: Twelfth symposium on usable privacy and security (SOUPS 2016) (pp 207–219). Denver, CO: USENIX Association. https://www.usenix.org/conference/soups2016/technical-sessions/presentation/krombholz. Accessed 22 July 2017

LeCun Y, Kavukcuoglu K, Farabet C (2010) Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE international symposium on circuits and systems (pp 253–256). Presented at the proceedings of 2010 IEEE international symposium on circuits and systems. https://doi.org/10.1109/ISCAS.2010.5537907

Lee S-H, Roh J-H, Kim S, Jin S-H (2016) A study on feature of keystroke dynamics for improving accuracy in mobile environment. In: Information security applications (pp 366–375). Presented at the international workshop on information security applications, Springer, Cham. https://doi.org/10.1007/978-3-319-56549-1_31

Li Y, Yang J, Xie M, Carlson D, Jang HG, Bian J (2015) Comparison of PIN- and pattern-based behavioral biometric authentication on mobile devices. In: MILCOM 2015–2015 IEEE Military communications conference (pp 1317–1322). Presented at the MILCOM 2015 –2015 IEEE military communications conference. https://doi.org/10.1109/MILCOM.2015.7357627

Martin A, Doddington G, Kamm T, Ordowski M, Przybocki M (1997) The DET curve in assessment of detection task performance. National Inst Of Standards And Technology Gaithersburg Md. http://www.dtic.mil/docs/citations/ADA530509. Accessed 11 Dec 2016

MathWorks (2016) Histogram bin counts—MATLAB histcounts. https://uk.mathworks.com/help/matlab/ref/histcounts.html?requestedDomain=www.mathworks.com. Accessed 11 Dec 2016

Miluzzo E, Cáceres R, Chen Y-F (2012) Vision: MClouds—computing on clouds of mobile devices. In: Proceedings of the third ACM workshop on mobile cloud computing and services (pp 9–14). New York, NY, USA: ACM. https://doi.org/10.1145/2307849.2307854

Owusu E, Han J, Das S, Perrig A, Zhang J (2012) ACCessory: password inference using accelerometers on smartphones. In: proceedings of the twelfth workshop on mobile computing systems & applications (pp 9:1–9:6). New York, NY, USA: ACM. https://doi.org/10.1145/2162081.2162095

Park YH, Tien DN, Lee HC, Park KR, Lee EC, Kim SM, Kim HC (2011) A multimodal biometric recognition of touched fingerprint and finger-vein. In: 2011 international conference on multimedia and signal processing (Vol. 1, pp 247–250). Presented at the 2011 international conference on multimedia and signal processing. https://doi.org/10.1109/CMSP.2011.57

Peng H, Long F, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238. https://doi.org/10.1109/TPAMI.2005.159

Pham XL, Nguyen TH, Chen GD (2017) Factors that impact quiz score: a study with participants in a mobile learning app. In: 2017 IEEE 17th international conference on advanced learning technologies (ICALT) (pp 103–105). Presented at the 2017 IEEE 17th international conference on advanced learning technologies (ICALT). https://doi.org/10.1109/ICALT.2017.81

Praher C, Sonntag M (2016) Applicability of keystroke dynamics as a biometric security feature for mobile touchscreen devices with virtualised keyboards. Int J Inf Comput Secur 8(1):72–91. https://doi.org/10.1504/IJICS.2016.075311

Prem SM (2016) Introductory statistics, 9th edn. Wiley, New York

Roh JH Lee SH, Kim S. (2016). Keystroke dynamics for authentication in smartphone. In: 2016 International Conference on Information and Communication Technology Convergence (ICTC) (pp 1155–1159). Presented at the 2016 International Conference on Information and Communication Technology Convergence (ICTC). https://doi.org/10.1109/ICTC.2016.7763394

Sen S, Muralidharan K (2014) Putting "pressure" on mobile authentication. In: 2014 seventh International Conference on mobile computing and ubiquitous networking (ICMU) (pp 56–61). Presented at the 2014 Seventh International Conference on Mobile Computing and Ubiquitous Networking (ICMU). https://doi.org/10.1109/ICMU.2014.6799058

Shen C, Yu T, Yuan S, Li Y, Guan X (2016) Performance analysis of motion-sensor behavior for user authentication on smartphones. Sensors 16(3):345. https://doi.org/10.3390/s16030345

Stanciu V-D, Spolaor R, Conti M, Giuffrida C (2016) On the effectiveness of sensor-enhanced keystroke dynamics against statistical attacks. In: proceedings of the sixth ACM conference on data and application security and privacy (pp 105–112). New York, NY, USA: ACM. https://doi.org/10.1145/2857705.2857748

Sun Y, Chen Y, Wang X, Tang X (2014) Deep learning face representation by joint identification-verification. In: Proceedings of the 27th international conference on neural information processing systems—vol 2 (pp 1988–1996). Cambridge, MA, USA: MIT Press. http://dl.acm.org/citation.cfm?id=2969033.2969049. Accessed 22 November 2017

Tasia C-J, Chang T-Y, Cheng P-C, Lin J-H (2014) Two novel biometric features in keystroke dynamics authentication systems for touch screen devices. Secur Commun Netw 7(4):750–758. https://doi.org/10.1002/sec.776

Tax DMJ (2001) One-class classification (Ph.D. thesis). Delft University of Technology. Retrieved from http://homepage.tudelft.nl/n9d04/thesis.pdf. Accessed 25 Apr 2016

Tax DMJ (2015) DDtools 2.1.2, the data description toolbox for matlab

Tax DMJ, Duin RPW (2004) Support vector data description. Mach Learn 54(1):45–66. https://doi.org/10.1023/B:MACH.0000008084.60811.49

Teh PS, Zhang N, Teoh ABJ, Chen K (2016a) A survey on touch dynamics authentication in mobile devices. Comput Secur 59:210–235. https://doi.org/10.1016/j.cose.2016.03.003

Teh PS, Zhang N, Teoh ABJ, Chen K (2016b) TDAS: a touch dynamics based multi-factor authentication solution for mobile devices. Int J Pervasive Comput Commun 12(1):127–153. https://doi.org/10.1108/IJPCC-01-2016-0005

Trojahn M, Arndt F, Ortmeier F (2013) Authentication with Keystroke dynamics on touchscreen keypads—effect of different N-graph combinations (pp 114–119). Presented at the MOBILITY 2013, The third international conference on mobile services, resources, and users. http://www.thinkmind.org/index.php?view=article&articleid=mobility_2013_5_30_40071. Accessed 17 Feb 2015

Wang J, Tang J, Xue G, Yang D (2017) Towards energy-efficient task scheduling on smartphones in mobile crowd sensing systems. Comput Netw 115:100–109. https://doi.org/10.1016/j.comnet.2016.11.020

Wu J, Chen Z (2015) An implicit identity authentication system considering changes of gesture based on keystroke behaviors. Int J Distrib Sens Netw 2015:e470274. https://doi.org/10.1155/2015/470274

Xu Y, Mo T, Feng Q, Zhong P, Lai, M, Chang EIC (2014) Deep learning of feature representation with multiple instance learning for medical image analysis. In *2014* IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp 1626–1630). Presented at the 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). https://doi.org/10.1109/ICASSP.2014.6853873

Zakaria N H, Griffiths D, Brostoff S, Yan J (2011) Shoulder surfing defence for recall-based graphical passwords. In: Proceedings of the seventh symposium on usable privacy and security (pp 6:1–6:12). New York, NY, USA: ACM. https://doi.org/10.1145/2078827.2078835

Zheng N, Bai K, Huang H, Wang H (2014) You are how you touch: user verification on smartphones via tapping behaviors. In: 2014 IEEE 22nd international conference on network protocols (ICNP) (pp 221–232). PRESENTED at the 2014 IEEE 22nd international conference on network protocols (ICNP). https://doi.org/10.1109/ICNP.2014.43